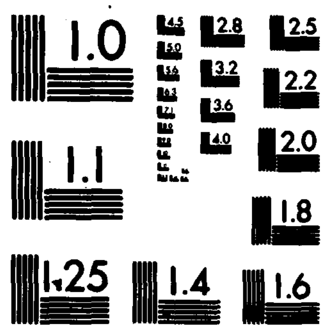
AD-A081 492 FLORIDA UNIV GAINESVILLE DEPT OF INDUSTRIAL AND SYS--ETC F/G 15/5
PRODUCTION LOT SIZING WITH MATERIAL HANDLING COST CONSIDERATION--ETC(U)
DEC 79 T J HODGSON, T J LOWE N00014-76-C-0096
UNCLASSIFIED RR-79-7 NL

END
DATE
FILMED
4-80



MICROCOPY RESOLUTION TEST CHART

12

6

PRODUCTION LOT SIZING WITH
MATERIAL HANDLING COST CONSIDERATIONS.

9 Research Rep[redacted]t, 79-7

by

10 T. J. Hodgson*
T. J. Lowe**

12 31

14 RR-79-7

11 Dec[redacted]79

*Department of Industrial and Systems Engineering
University of Florida
Gainesville, Florida 32611

**Krannert Graduate School of Management
Purdue University
West Lafayette, Ind. 47907

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

This research was supported in part by the Office of Naval
Research, under contract number N00014-76-C-0096.
15

THE FINDINGS OF THIS REPORT ARE NOT TO BE CONSTRUED AS AN
OFFICIAL DEPARTMENT OF THE NAVY POSITION, UNLESS SO DESIGNATED
BY OTHER AUTHORIZED DOCUMENTS.

404399

mt

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 79-7	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) Production Lot Sizing With Material Handling Cost Considerations	5. TYPE OF REPORT & PERIOD COVERED Technical
	6. PERFORMING ORG. REPORT NUMBER 79-7
7. AUTHOR(s) T. J. Hodgson and T. J. Lowe	8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0096
9. PERFORMING ORGANIZATION NAME AND ADDRESS Industrial and Systems Engineering Department University of Florida Gainesville, Florida 32611	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2Q061102A14D/ Rsch in & Appl of Applied Math.
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA	12. REPORT DATE December, 1979
	13. NUMBER OF PAGES 30
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

N/A

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Lot Sizing, Material Handling, Automated Warehousing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The determination of production lot sizes and the assignment of storage space in a warehouse for the produced items are usually treated as two separate problems: the former providing input (space needed) to the latter. In this paper, we treat the decision problems as one with the objective of minimizing total setup, inventory carrying, and warehouse material handling cost. We treat the minimum material handling cost as a continuous function of the lot sizes and develop an algorithm for finding locally optimal solutions of the derived optimization problem. Computational experience is provided and applications to automated warehousing systems are discussed.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

Abstract

The determination of production lot sizes and the assignment of storage space in a warehouse for the produced items are usually treated as two separate problems: The former providing input (space needed) to the latter. In this paper, we treat the decision problems as one with the objective of minimizing total setup, inventory carrying, and warehouse material handling cost. We treat the minimum material handling cost as a continuous function of the lot sizes and develop an algorithm for finding locally optimal solutions of the derived optimization problem. Computational experience is provided and applications to automated warehousing systems are discussed.

1. Introduction

In any automatic warehousing system utilizing computer-controlled stacker cranes for high volume material handling, a rather broad class of decision problems can be identified. The function of the system is the automatic control of the storage and retrieval of items in a warehouse. As is briefly described in [4], and [6], incoming items are assigned to pallets, perhaps several items to a pallet, and then a minicomputer assigns the pallet to a location in the storage racks. The pallet contents and its location are recorded in computer memory. The material handling is done by automatic stacker cranes.

When a request for an item in storage is received, the computer obtains the pallet location address from computer memory and the stacker crane retrieves the pallet. Once the material is removed from the pallet, the pallet may be stored for future use if it is empty, or returned to the storage location if it still contains items.

Decision problems associated with automatic warehousing systems are classified in [6] as <u>design</u> and <u>scheduling</u>. Design questions involve such things as the number of stacker cranes to be used, the physical dimensions of the storage bays, the number of storage bays, and perhaps even the physical dimensions of the warehouse.

Scheduling questions involve pallet assignment (assigning items to pallets), storage assignment (assigning loaded pallets to storage locations), and interleaving (the sequencing rules for storage and retrieve requests). It is clear that design and scheduling questions should be addressed simultaneously to obtain the most benefit from the automatic warehousing system.

Several recent papers [4], [6], [7] have addressed the storage assignment problem, with the objective of assigning items to storage locations in order to minimize total material handling time or cost. In [6] several storage assignment rules are compared under the assumption of no interleaving. In [4], the analysis is extended to the case with interleaving. Heskett [5] proposed that inventory items should be assigned to storage locations according to the cube order index (COI) rule. The COI is the ratio of the space required per item and the order frequency of the item. An application of the use of the COI rule can be found in [7]. It has been shown [3], that if a certain "factoring assumption" holds, the COI rule will minimize total material handling cost (or time). An underlying assumption in the above references is that the quantities of each particular class of items to be stored is known apriori.

In a production setting, the amount of product warehouse space needed for a particular item is traditionally determined on the basis of inventory and production cost considerations. As an example, if a company is using economic order quantity analysis to determine production lot sizes, then the production quantity decisions resulting from this analysis will determine the amount of warehouse space which must be allocated to each product. Thus, traditionally, the production lot size decisions and the warehouse allocation decisions are treated independently, the former providing information (space needed) to the latter.

This paper concerns treating these two decision problems as one. That is, to analyze the problem of determining order quantities for several products and decision rules for allocating warehouse space for these products in order to minimize total production cost, inventory carrying cost and material handling cost. Thus, in a sense we are expanding the definition of "scheduling" as given in [4] and [6] to include the simultaneous determination of production lot sizes and storage locations for final products.

In a recent paper, Wilson [13] addressed the above problem, but treated the allocation problem as a discrete problem. That is, he assumed a finite number of locations in the warehouse and treated the space allocation problem as an assignment problem. In treating the space allocation problem in this fashion, a problem with a large number of warehouse locations becomes difficult to solve [13]. In this paper, we treat the space allocation problem as a continuous layout problem [3]. This is the approach taken in [4], [6], [10] and [12].

There is little loss in using the continuous approach. The vast majority of automated warehouses which could benefit from this analysis are large enough that "rounding off" a continuous solution to fit the actual discrete locations in the warehouse would not affect the actual solution objective function greatly. The continuous approach has the important advantage of relative ease of solution.

Consider a production vector $(Q_1, Q_2, \ldots, Q_n)$, where Q_i represents the production lot size of product i (we suppose that the company produces n different products). The total yearly production, inventory carrying, and material handling cost for this production vector can be expressed as:

$$T(Q_1, \ldots, Q_n) = \sum_{i=1}^{n} P_i(Q_i) + \sum_{i=1}^{n} I_i(Q_i) + M(Q_1, \ldots, Q_n), \tag{1}$$

where $P_i(Q_i)$ represents the yearly production cost of product i (as a function of Q_i), $I_i(Q_i)$ represents the yearly inventory carrying cost of product i (as a function of Q_i), and $M(Q_1, \ldots, Q_n)$ represents the minimum yearly material handling cost as a function of the production vector. That is, given the production vector, M represents the material handling cost which results from an optimal (cost minimizing) storage scheme. Note that we have treated the production and inventory carrying costs as separable by product. In general, this is not a bad assumption if sufficient production capacity is available. However, the yearly material handling cost is not separable since the material handling cost for product i depends not only on Q_i, but also on all Q_j, $j \neq i$. This dependence is easily seen by noting that material handling cost arises from the storage and retrieval of items in a warehouse. The cost for movement of item i depends upon the amount of warehouse space required for items other than item i.

Under the classical derivation, the functions $P_i(Q_i)$ and $I_i(Q_i)$ are convex functions of their arguments. In general, the function $M(Q_1, \ldots, Q_n)$ is not a convex function of $(Q_1, \ldots, Q_n)$, but by treating the space allocation problem as a continuous layout problem, and assuming that material handling cost is a linear function of the distance between item location in the warehouse and the warehouse input/output (I/O) point (loading dock), we can derive a closed form expression for $M(Q_1, \ldots, Q_n)$, once an "ordering" relation between the Q_i is determined. For a fixed ordering of the Q_i, $M(Q_1, \ldots, Q_n)$ is differentiable.

The main advantage of treating the problem as formulated in (1) is that the problem size is drastically reduced over the formulation as given by Wilson. In Wilson's formulation if there are m discrete storage locations, then the number of variables is m times the number of products. Under the formulation in (1), the number of variables is equal to the number of products.

At this point we give a brief overview of the paper. In Section 2 we outline the assumptions and notation used in later sections. In Section 3 we analyze the underlying problem when the crane travel is assumed to be rectilinear or Tchebychev [12], and formulate (1) for this case. We provide an algorithm to find locally optimal solutions to the resulting problem and give computational results. In Section 4 we examine a special case of the problem formulated in Section 3 and relate the resulting problem to a single processor scheduling problem. Globally optimal solutions for this problem can be efficiently obtained for medium sized problems ($n \leq 25$).

2. Assumptions and Notation

As mentioned in the previous section, we will treat the storage assignment problem as a continuous layout problem. We assume that during each storage trip from the I/O point to the storage area, the stacker crane will carry only one item. Further, during any retrieve trip, only one item will be transferred from the storage area to the I/O point. We also assume that the material handling system operates with no interleaving. This basically means that the crane returns to the I/O point between simultaneous move requests (either store or retrieve). The material handling cost for an individual trip to or from the storage area will be a linear function of the time involved in the trip. We allow for the possibility that each item may have its own linear function. Let t_i be the cost, per unit of crane travel time, for item i. The actual time for material loading or unloading is ignored. As is pointed out in [6], these times are small compared to crane travel time. We analyze a system which consists of a single crane and one storage rack (area). It is easiest to visualize the storage area as a plane oriented in the vertical direction (see Figure 1). The analysis is easily extendable to the case of a single crane serving a single two sided aisle, however this case unnecessarily complicates the analysis.

In this paper we assume that the production cost function $P_i(Q_i)$ is of the classical form. That is, if D_i is the yearly demand rate for item i and if C_i is the setup (or order cost) associated with each lot of item i, then $P_i(Q_i) = C_i\ D_i/Q_i$. We remark that more complex production cost functions can be handled. Similarily, we could consider more complex inventory carrying cost functions, but we assume that $I_i(Q_i) = k_i Q_i/2$, where the coefficient k_i reflects the cost of item i, the company's cost of capital, etc. Since we will be dealing with storage area for each of the items, let A_i represent the area necessary to store one unit of item i.

The form of the function $M(Q_1, \ldots, Q_n)$ in (1) will be dependent upon the crane travel assumptions, and the physical dimensions of the storage area. In Section 3 we analyze the problem under the assumption that the crane can travel, simultaneously, both horizontally and vertically.

3. Tchebychev Crane Travel

In this section, we assume that the I/O point is located in a corner of the storage area (as shown in Figure 1). Further, assume that the crane can travel, simultaneously, in both the horizontal and vertical directions. Let (0,0) be the coordinates of the location of the I/O point, and any point in the storage area has coordinates (x,y), where $x \geq 0$, $y \geq 0$ (See Figure 1).

Let v_1 and v_2 be the horizontal and vertical speed, respectively, of the crane. If an item is stored at location (x,y) in the storage area, the total time for the crane to travel between the I/O point and the location of the item is $\max\{x/v_1, y/v_2\}$, and the total crane travel time for storage, and eventual retrieval of this item is $4 \max\{x/v_1, y/v_2\}$, where the number 4 reflects two round trips between the I/O point and the location of the item.

To obtain insight into the problem, we will initially formulate expression (1) for the single item case; thus, suppose n=1. For a fixed value of Q_1, it can be shown (see [3]), that under the assumption that the material handling cost is linear in travel time, the optimal storage layout for the single item is a region, of area A_1Q_1, enclosed by a contour of the function $f(x,y) = \max\{x/v_1, y/v_2\}$. Following the development in [3], let z be the functional value of a contour of f. The area, q(z) of the optimal storage region can be expressed as a function of time by $q(z) = v_1v_2z^2$. Since the total area of the storage region must equal A_1Q_1, set $A_1Q_1 = v_1v_2z^2$. Thus, the value of the function f at the boundary of the storage region is $z = (A_1Q_1/v_1v_2)^{\frac{1}{2}}$.

The average time for the crane to travel from (to) the I/O point to (from) the location of one unit of the item in the specified storage region is (see [3])

$$(1/A_1Q_1)\int_0^{(A_1Q_1/v_1v_2)^{\frac{1}{2}}} (2v_1v_2z)z \; dz = (2/3)(A_1Q_1/v_1v_2)^{\frac{1}{2}}.$$

Since for each item the crane makes 4 one-way trips between the location of the item and the I/O point, and since D_1 items are "turned over" in the storage area per year, it follows that the total minimum yearly material handling cost, $M(Q_1)$ is given by

$$M(Q_1) = (8t_1D_1/3)(A_1Q_1/v_1v_2)^{\frac{1}{2}}.$$

Prior to formulating and analyzing the n item case, we demonstrate that the minimization of (1) for the case n=1 over $Q_1 \geqq 0$ is relatively straightforward.

Although $P_1(Q_1) = C_1D_1/Q_1$ and $I_1(Q_1) = k_1Q_1/2$ are convex in Q_1, it is clear that $M(Q_1)$ is not convex in Q_1. However, we show that there exists a single local minimum of $T(Q_1)$ over $Q_1 \geqq 0$ and thus any of a variety of one dimensional search procedures (see [9] or [14] will minimize $T(Q_1)$.

Let $\beta_1 = C_1D_1$, $\beta_2 = k_1/2$, and $\beta_3 = (8t_1D_1/3)(A_1/v_1v_2)^{\frac{1}{2}}$, where clearly $\beta_i > 0$, $i = 1, 2, 3$. Thus,

$$T(Q_1) = \beta_1/Q_1 + \beta_2Q_1 + \beta_3Q_1^{\frac{1}{2}}. \tag{2}$$

<u>Property 1</u> There exists a single local minimum, Q_1^*, of $T(Q_1)$ on $\{Q_1 | Q_1 \geqq 0\}$.

<u>Proof</u>: Since $T(Q_1) \to \infty$ as either $Q_1 \to 0$ or $Q_1 \to +\infty$, and since T is continuous and is finite for any positive finite Q_1, it follows that there exists at least one positive finite local minimum of T.

Since T is differentiable everywhere on $\{Q_1 | Q_1 > 0\}$, $T' = 0$ at any local minimum of T. Differentiating (2) gives

$$T'(Q_1) = -\beta_1/Q_1^2 + \beta_2 + \beta_3Q_1^{-\frac{1}{2}}/2. \tag{3}$$

For any $\hat{Q}_1 > 0$, $T'(\hat{Q}_1) = 0$ if and only if $\hat{Q}_1 T'(\hat{Q}_1) = 0$. Thus, multiplying (3) by Q_1 and setting the result equal to zero gives $Q_1 T'(Q_1) = -\beta_1/Q_1 + \beta_2 Q_1 + \beta_3 Q_1^{\frac{1}{2}}/2 = 0$. Since $\beta_i > 0$, $i = 1,2,3$, $Q_1 T'(Q_1) < 0$ for sufficiently small $Q_1 > 0$ and $Q_1 T'(Q_1) > 0$ for sufficiently large $Q_1 > 0$. Furthermore, $Q_1 T'(Q_1)$ is strictly increasing on $\{Q_1 | Q_1 > 0\}$. Since $Q_1 T'(Q_1)$ is continuous, it follows that there exists exactly one point $Q_1^* > 0$ such that $Q_1^* T'(Q_1^*) = 0$ and hence $T'(Q_1^*) = 0$.

We remark that $T(Q_1)$ is psuedo-convex on $\{Q_1 | Q_1 > 0\}$ since (2) can be written as $\phi(Q_1)/Q_1$, where $\phi(Q_1)$ is positive and convex on $\{Q_1 | Q_1 > 0\}$ (see [1], pp. 154-156).

We now formulate the n item case where we rely heavily on the notation and development in [3]. Let $L = \{(x, y) \mid x \geqq 0, y \geqq 0\}$, i.e., L denotes the set of points in the storage rack. As before, the travel time function to any point $(x,y) \in L$ is expressed by $f(x,y) = \max\{x/v_1, y/v_2\}$. We now formulate a closed form expression for $M(Q_1, \ldots, Q_n)$ at any <u>fixed</u> $(Q_1, \ldots, Q_n)$. Let $\{S_i, i = 1, \ldots, n\}$ be a collection of nonoverlapping subsets of L (a <u>layout</u>) such that the area of $S_i = A_i Q_i$, $i = 1, \ldots, n$. $H_n(L,A)$ denotes the collection of all such layouts. If item i is assigned storage region S_i, the <u>average</u> one way travel time between the I/O point and the location of an item i is

$$(1/A_i Q_i) \int_{S_i} f \quad .$$

The yearly material handling cost for item i is

$$(4t_i D_i/A_i Q_i) \int_{S_i} f \;, \tag{4}$$

where (4) reflects 2 round trips for each item, D_i items per year, and a cost of t_i per unit time. For notational convenience in what follows, define $r_i = 4t_i D_i / A_i Q_i$. Using (11), $M(Q_1, \ldots, Q_n)$ becomes

$$M(Q_1, \ldots, Q_n) = \min_{(S_1, \ldots, S_n) \in H_n(L,A)} \left[\sum_{i=1}^{n} r_i \int_{S_i} f \right]. \qquad (5)$$

Due to the analysis in [3], the solution $\{S_i^*, i=1, \ldots, n\}$, to the right hand side of (5) can be found by:

i) Ordering the items such that

$$r_{[i]} \geq r_{[i+1]}, \quad i=1, \ldots, n-1. \qquad (6)$$

ii) Defining $S^*_{[1]}$ as the region of L closest to the I/O point, enclosed by a contour of f; defining $S^*_{[2]}$ as the region nesting about $S^*_{[1]}$ and enclosed by another (larger valued) contour of f; and so on (see Figure 2).

Using ii) and the definition of f, $M(Q_1, \ldots, Q_n)$ can be found by integrating the right hand side of (5):

$$M(Q_1, \ldots, Q_n) = (2/3(v_1 v_2)^{\frac{1}{2}}) \sum_{i=1}^{n} r_{[i]} \left[\left(\sum_{j=1}^{i} A_{[j]} Q_{[j]}\right)^{3/2} - \left(\sum_{j=1}^{i-1} A_{[j]} Q_{[j]}\right)^{3/2} \right]. \qquad (7)$$

When $i = 1$ in (7), we define $\sum_{j=1}^{i-1} A_{[j]} Q_{[j]} = 0$ and so for the single item case, (7) agrees with our earlier analysis. We emphasize the fact that the ordering $[1], \ldots, [n]$ in (7) is <u>dependent</u> upon the values $Q_1, \ldots, Q_n$ and so the functional form of M is known only after $Q_1, \ldots, Q_n$ are known.

Using (7) in (1) along with the assumed form of $P_i(Q_i)$ and $I_i(Q_i)$, no apparent conclusion can be made about the properties of $T(Q_1, \ldots, Q_n)$. Even though $P_i(Q_i)$ and $I_i(Q_i)$ are convex, it is not clear that the right hand side of (7) even for a fixed ordering [1], ..., [n] of the items has any desirable properties (such as psuedo-convexity) which could be exploited in optimizing $T(Q_1, \ldots, Q_n)$. However, for a fixed ordering [1], ..., [n] of the items, the function generated by adding the right hand side of (7) to

$$\sum_{i=1}^{n} P_i(Q_i) + I_i(Q_i), \qquad (8)$$

is differentiable, and hence at any local minimum the gradient vanishes. We exploit this property in the following algorithm. We choose initial positive values $Q_1^1, \ldots, Q_n^1$ and order the items as [1], ..., [n] such that (6) holds at $Q_1^1, \ldots, Q_n^1$. We then form the function defined by adding the right hand side of (7) to (8) and find a local minimum of this function. If (6) holds at this local minimum with the initial ordering, stop. Otherwise reorder the items so that (6) holds (forming a new function) and find a local minimum of the new function, etc.

Formally, at some iterate $Q^k = \{Q_1^k, \ldots, Q_n^k\}$, let ρ^k be an ordering of the items such that (6) holds and let T^k be the function defined by adding (8) to the right hand side of (7) with the ordering induced by ρ^k. Let T^{k*} be the function value of T^k at a local minimum of T^k. Define as $Q^{k+1} = \{Q_1^{k+1}, \ldots, Q_n^{k+1}\}$ the vector which attains T^{k*}.

Step 1 Choose an initial starting point $Q^1 > 0$. (A natural starting point is some fraction of the solution obtained by minimizing (8). Set $k = 1$ and determine ρ^k.

Step 2 Form T^k.

Step 3 Find a T^{k*} and Q^{k+1}.

Step 4 Determine ρ^{k+1}. If $\rho^{k+1} = \rho^k$, stop. Otherwise, set $k = k + 1$ and go to Step 2.

In the above algorithm, at some iterate Q^k, it may be the case that more than one ordering of the items satisfies (6). In this case we adopt the convention that ρ^k is <u>any</u> ordering of the items which satisfies (6). In Step 4, by $\rho^{k+1} = \rho^k$ we mean that if $[i]$, $i = 1, \ldots,$ is the ordering of the items induced by ρ^k, then at Q^{k+1}, $D_{[i]}t_{[i]}/A_{[i]}Q^{k+1}_{[i]} \geq D_{[i+1]}t_{[i+1]}/A_{[i+1]}Q^{k+1}_{[i+1]}$, $i=1, \ldots, n-1$.

A property of the algorithm is that the objective function strictly decreases at each transition from Step 4 to Step 2 (reordering of items). The reason for this is that $\rho^{k+1} = \rho^k$ is a necessary condition for optimality. We establish this fact in what follows.

Consider any $Q = \{Q_1, \ldots, Q_n\}$, where $Q_i > 0$, $i = 1, \ldots, n$. In ii) let $[i] = i$, $i = 1, \ldots, n$ and let $(\bar{S}_1, \ldots, \bar{S}_n)$ be the member of $H_n(L,A)$ defined by ii). That is, the sets $\bar{S}_i$ are enclosed by contours of f and $\bar{S}_{i+1}$ nests about $\bar{S}_i$, $i = 1, \ldots, n-1$.

Suppose that for some $j \leq n-1$, $r_j < r_{j+1}$. Let $(\hat{S}_1, \ldots, \hat{S}_n)$ be the member of $H_n(L,A)$ where $\hat{S}_\ell = \bar{S}_\ell$ for all $\ell \neq j$ or $j+1$, $\hat{S}_{j+1}$ nests about $\hat{S}_{j-1}$, $\hat{S}_j$ nests about $\hat{S}_{j+1}$, and $\hat{S}_{j+1}$ and $\hat{S}_j$ are enclosed by contours of f. (Generating $\{\hat{S}_1, \ldots, \hat{S}_n\}$ from $\{\bar{S}_1, \ldots, \bar{S}_n\}$ can be thought of as leaving $\bar{S}_\ell$ as is for all $\ell \neq j$ or $j+1$; finding the contour of f such that the region enclosed by f, surrounding $\bigcup_{\ell=1}^{j-1} \hat{S}_\ell \left(= \bigcup_{\ell=1}^{j-1} \bar{S}_\ell\right)$, is of area $= A_{j+1}Q_{j+1}$ and denoting this region as $\hat{S}_{j+1}$. $\hat{S}_j$ is then the region surrounding $\hat{S}_{j+1}$ and surrounded by $\bigcup_{\ell=j+2}^{n} \hat{S}_\ell$.) Given this construct, we now establish a result which leads to a necessary condition for optimality.

Property 2

$$\sum_{i=1}^{n} r_i \int_{\bar{S}_i} f - \sum_{i=1}^{n} r_i \int_{\hat{S}_i} f > 0. \tag{9}$$

Proof: Since $\bar{S}_\ell = \hat{S}_\ell$, $\ell \neq j$ or $j+1$, the left hand side of (9) becomes

$$r_j \int_{\bar{S}_j} f + r_{j+1} \int_{\bar{S}_{j+1}} f - r_j \int_{\hat{S}_j} f - r_{j+1} \int_{\hat{S}_{j+1}} f \quad . \tag{10}$$

Without loss of generality, suppose that $A_j Q_j \geq A_{j+1} Q_{j+1}$.

Adding

$$r_j \int_{\bar{S}_{j+1}} f - r_j \int_{\bar{S}_{j+1}} f + r_j \int_{\hat{S}_{j+1}} f - r_j \int_{\hat{S}_{j+1}} f \quad ,$$

which equals zero, to (10) gives

$$(r_{j+1} - r_j) \int_{\bar{S}_{j+1}} f - (r_{j+1} - r_j) \int_{\hat{S}_{j+1}} f + r_j \int_{\bar{S}_j \cup \bar{S}_{j+1}} f - r_j \int_{\hat{S}_j \cup \hat{S}_{j+1}} f \quad . \tag{11}$$

Since $\bar{S}_j \cup \bar{S}_{j+1} = \hat{S}_j \cup \hat{S}_{j+1}$, (11) is equivalent to

$$(r_{j+1} - r_j)\left[\int_{\bar{S}_{j+1}} f - \int_{\hat{S}_{j+1}} f\right]. \tag{12}$$

Due to the fact that f is strictly increasing along any ray in L originating at (0,0) and further that the area of $\bar{S}_j$ = area of $\hat{S}_j$, area of $\bar{S}_{j+1}$ = area of $\hat{S}_{j+1}$, and area of $\hat{S}_{j+1} \geq$ area of $\hat{S}_j$, it follows that for arbitrary $(x,y) \in \bar{S}_{j+1}$ and arbitrary $(x',y') \in \hat{S}_{j+1}$,

$$f(x,y) \geq f(x',y'). \tag{13}$$

Furthermore, there exist subsets of equal positive area in $\bar{S}_{j+1}$ and $\hat{S}_{j+1}$ over which (13) holds at strict inequality. Thus the term in brackets in (12) is strictly positive. Since $r_{j+1} - r_j > 0$, (12) is strictly greater than zero, establishing (9).

Property 2 is the key to the following result which establishes that the heuristic is strictly decreasing in each iteration.

Property 3 If $\rho^{k} \neq \rho^{k+1}$, then

$$T^{k*} - T^{k+1*} > 0. \tag{14}$$

Proof: By definition, $T^{k*} = T^{k}(Q^{k+1})$. In addition, it follows that $T^{k+1}(Q^{k+1}) \geqq T^{k+1*}$, and thus to establish (14), it is sufficient to show that

$$T^{k}(Q^{k+1}) - T^{k+1}(Q^{k+1}) > 0. \tag{15}$$

Without loss of generality, suppose that ρ^{k} induces the ordering $[i] = i$, $i = 1, \ldots, n$. Since $\rho^{k} \neq \rho^{k+1}$, there exists some $j \leqq n-1$ such that $r_j < r_{j+1}$ where the r terms are evaluated at Q^{k+1}. Suppose that there is only one such j with this property. Thus the ordering ρ^{k+1} can be obtained by interchanging the order of j and j+1.

Due to the form of T^{k} and T^{k+1},

$$T^{k}(Q^{k+1}) - T^{k+1}(Q^{k+1}) = \sum_{i=1}^{n} r_i \int_{\bar{S}_i} f - \sum_{i=1}^{n} r_i \int_{\hat{S}_i} f, \tag{16}$$

where $\{\bar{S}_i\}$ and $\{\hat{S}_i\}$ are as in Property 2. But, by Property 2, the right hand side of (16) is strictly greater than zero. Thus (15) and hence (14) holds.

In the case where there is more than one $j \leqq n-1$ where $r_j < r_{j+1}$, the ordering ρ^{k+1} can be obtained by making pairwise interchanges of "adjacent" items where this property holds. Each such interchange strictly decreases the material handling cost associated with the generated layout. Thus (14) holds in this case as well.

A FORTRAN program implementing the algorithm has been written and tested on the AMDAHL 470 computer. In the algorithm, Newton's method [14] was used in Step 3. Fifty-eight problems with $n = 5$ were run with an average computation time of .02 seconds per problem. In these problems $A_i = 1$ for all i; C_i, D_i, k_i and t_i were randomly generated. Seventeen of these problems required at least one reordering of the items (transition from Step 4 to Step 2) prior to termination. In one problem (.03 seconds) the items were reordered 4 times and the total cost was reduced by one half (initial solution was evaluated at the EOQ values for the items). To test computation time sensitivity to the size of n, problems of various dimensionality were run. The results of these tests appear in Table 1.

4. Travel Dominance

In this section, we formulate and analyze a special case of the model presented in Section 3. We also discuss procedures to find the optimal solution which are computationally feasible for medium sized problems.

As in Section 3, suppose that the crane travel is Tchebychev with horizontal and vertical speeds of v_1 and v_2, respectively. As before, let L denote the set of points in the storage rack. Given any point $(x,y) \in L$, let $d_1(x,y)$ and $d_2(x,y)$ be the horizontal and vertical distances, respectively, from (x,y) to the I/O point. The basic assumption in this section is that either:

$$d_1(x,y)/v_1 \geq d_2(x,y)/v_2 \text{ for all } (x,y) \in L, \tag{17}$$

$$\text{or } d_2(x,y)/v_2 \geq d_1(x,y)/v_1 \text{ for all } (x,y) \in L. \tag{18}$$

The essence of (17) or (18) above is that either the horizontal travel time dominants the vertical travel time or vice versa over every location in the storage rack. In this section, we assume that (17) holds. This situation could occur for several reasons. First, suppose the I/O point for the crane is located at the end of a conveyor belt which is not adjacent to the storage rack but instead is of horizontal distance W from the storage rack. Further, suppose that the height of the storage rack is H (see Figure 3). Then, if simultaneous horizontal and vertical travel is allowed between the I/O point and the storage rack, and if $W/v_1 \geq H/v_2$, then (17) always holds. This is due to the fact that for any $(x,y) \in L$, $d_1(x,y) \geq W$ and $d_2(x,y) \leq H$ and thus $d_1(x,y)/v_1 \geq W/v_1 \geq H/v_2 \geq d_2(x,y)/v_2$ establishing (17). In another situation, it may be that the horizontal speed of the crane, v_1, is much slower than the vertical speed, v_2. In this case $d_1(x,y)/v_1$ "approximates" the travel time to any point in L.

We now formulate the model assuming that (17) holds. Referring to Figure 3, the I/O point has coordinates (0,0). Any point $(x,y) \in L$ has coordinates $W \leq x$ and $0 \leq y \leq H$. Due to (17), $x/v_1 \geq y/v_2$ and so the one way travel time between the I/O point and $(x,y) \in L$ is $f(x,y) = x/v_1$. Noting that the contours of f are vertical lines, it follows that the solution $(S_1^*, S_2^*, \ldots, S_n^*)$ to the right hand side of (5) for a given $(Q_1, \ldots, Q_n)$ can be found by ordering the items such that (6) holds, and defining

$$S_{[1]}^* = \{(x,y) \in L \mid W \leq x \leq W + (A_{[1]}Q_{[1]})/H,\ 0 \leq y \leq H\}, \text{ and}$$

$$S_{[j]}^* = \{(x,y) \in L \mid W + \sum_{i=1}^{j-1} A_{[i]}Q_{[i]}/H \leq x \leq W + \sum_{i=1}^{j} A_{[i]}Q_{[i]}/H,\ 0 \leq y \leq H\},$$

$$j = 2, \ldots, n. \tag{19}$$

A closed form expression for $M(Q_1, \ldots, Q_n)$ can be found by using (19) in (5), carrying out the integration in (5), and rearranging terms:

$$\begin{aligned} M(Q_{[1]}, \ldots, Q_{[n]}) = {} & \sum_{i=1}^{n} 4Wt_{[i]}D_{[i]}/v_1 \\ & + (Q_{[1]}A_{[1]})(4t_{[1]}D_{[1]} + 8t_{[2]}D_{[2]} + \cdots + 8t_{[n]}D_{[n]})/2v_1H \\ & + (Q_{[2]}A_{[2]})(4t_{[2]}D_{[2]} + 8t_{[3]}D_{[3]} + \cdots + 8t_{[n]}D_{[n]})/2v_1H \\ & + \\ & \cdot \\ & \cdot \\ & + (Q_{[n]}A_{[n]})(4t_{[n]}D_{[n]})/2v_1H, \end{aligned} \tag{20}$$

with the understanding that (6) holds.

We now formulate (1) for this travel dominance case with $P_i(Q_i)$ and $I_i(Q_i)$ as before. Using (20), we obtain, upon rearranging

$$\begin{aligned} T(Q_1, \ldots, Q_n) = {} & \sum_{i=1}^{n} 4Wt_{[i]}D_{[i]}/v_1 + \sum_{i=1}^{n} C_{[i]}D_{[i]}/Q_{[i]} \\ & + \sum_{i=1}^{n} (Q_{[i]}/2)(k_{[i]} + (A_{[i]}/v_1H)(4t_{[i]}D_{[i]} + 8(\sum_{j=i+1}^{n} t_{[j]}D_{[j]}))) \end{aligned} \tag{21}$$

We emphasize that the right hand side of (21) is a valid description of T at $(Q_1, \ldots, Q_n)$, $Q_j > 0$, $j = 1, \ldots, n$, if only if the items are indexed such that (6) holds. Note that the first term in (21) is constant for any ordering of the items and thus the "⌈ ⌉" notation can be dropped from this term. Also, for any <u>fixed</u> ordering $\lceil 1 \rceil, \lceil 2 \rceil, \ldots, \lceil n \rceil$ of items, the right hand side of (21) is convex, differentiable and separable. Further, for a <u>fixed</u> ordering of the items, the values of $Q^*_{\lceil j \rceil}$, $j = 1, \ldots, n$ that minimize the right hand side of (21) are easily obtained from the now classical EOQ formulas.

The above discussion suggests a procedure for finding the minimum of $T(Q_1, \ldots, Q_n)$. Let $P = \{P_1, P_2, \ldots, P_n\}$ be a permutation of the integers $1, 2, \ldots, n$. Let $(Q^*_{P_1}, \ldots, Q^*_{P_n})$ be the values of $(Q_{P_1}, \ldots, Q_{P_n})$ which minimize the right hand side of (21), where P_i replaces $\lceil i \rceil$ in (21). Define T^*_P as the value of the right hand side of (21) at $(Q^*_{P_1}, \ldots, Q^*_{P_n})$. Then the minimum of $T(Q_1, \ldots, Q_n)$ can be found by solving

$$\min_P T^*_P$$

$$\text{s.t.}$$

$$r_{P_j} > r_{P_{j+1}}, \quad j = 1, \ldots, n, \tag{22}$$

where r_{P_k} is computed at $Q^*_{P_k}$.

Thus, to find the minimum of $T(Q_1, \ldots, Q_n)$, we could generate every permutation $P = \{P_1, \ldots, P_n\}$ of the integers $1, \ldots, n$, find T^*_P for all P, and choose from among those permutations P where (22) holds, the permutation yielding the minimum T^*_P. However, it follows from Property 3 that (22) must hold at $(Q^*_{P_1}, \ldots, Q^*_{P_n})$ if P is the optimal permutation. To see this, it follows from Property 3 that if (22) does not hold for some $j \leq n-1$, the ordering of j and j+1 can be interchanged thereby decreasing the material handling portion of T. Thus it is only necessary to compute T_P^* for all permutations P and choose the smallest.

For a fixed permutation P, T_P^* and $(Q_{P_1}^*, \ldots, Q_{P_n}^*)$ are easily obtained for the following formulas:

$$Q_{P_i}^* = \left[2C_{P_i}D_{P_i}/(k_{P_i} + (A_{P_i}/v_1 H)(4t_{P_i}D_{P_i} + 8(\sum_{j=i+1}^{n} t_{P_j}D_{P_j})))\right]^{\frac{1}{2}}, \quad i=1, \ldots, n. \tag{23}$$

$$T_P^* = \sum_{i=1}^{n} 4Wt_i D_i/v_1 + \sum_{i=1}^{n} \left[2C_{P_i}D_{P_i}(k_{P_i} + (A_{P_i}/v_1 H)(4t_{P_i}D_{P_i} + 8(\sum_{j=i+1}^{n} t_{P_j}D_{P_j})))\right]^{\frac{1}{2}}. \tag{24}$$

Note that (23) and (24) are closed form expressions in terms of the data. Thus, solving the problem involves computing (24) for every permutation, P, selecting a permutation which gives the minimum T_P^* and then choosing the optimal item quantities according to (23). Obviously, this procedure involves O(n!) computations, which is computationally feasible for relatively small problems.

We now outline a computational procedure which involves somewhat less work than the above. The above problem can be interpreted as single processor scheduling problem with monotonic increasing deferral costs. Pursuant to this goal, for a given permutation, P, define for i=1, ..., n,

$\gamma_{P_i} \equiv t_{P_i}D_{P_i}$; $\alpha_{P_i} \equiv 2C_{P_i}D_{P_i}(k_{P_i} - (4A_{P_i}\gamma_{P_i})/v_1 H))$; and $\beta_{P_i} \equiv 16C_{P_i}D_{P_i}A_{P_i}/v_1 H$.

With these definitions, (24) becomes

$$T_P^* = K + \sum_{i=1}^{n} [\alpha_{P_i} + \beta_{P_i}(\sum_{j=i}^{n} \gamma_{P_j})]^{\frac{1}{2}}, \tag{25}$$

where K is the constant $\sum_{i=1}^{n} 4W\gamma_{P_i}/v_1$. The interpretation of (25) as a scheduling problem is relatively straightforward. v_{P_i} represents the "processing time" of job P_i. If job P_n is scheduled first on the machine, job P_{n-1} second, ..., and job P_1 last, then $\sum_{j=i}^{n} \gamma_{P_j}$ represents the "flow time" of the job which is scheduled in position n-i+1 on the machine. β_{P_i} represents a unit penalty cost of flow time. We can think of α_{P_i} as representing "off-machine" time (perhaps time to package the job after it is completed). Finally the square root function represents the increasing deferral cost function (as a function total time in the shop).

Baker [2] and Lawler [8] have outlined a dynamic programming procedure to solve very general single machine sequencing problems. Since the deferral costs are monotone nondecreasing and (25) is additive, the dynamic programming procedure can be applied. Lawler [8] points out that the computational effort of the dynamic programming procedure is $O(n2^n)$ and that problems of size $n \leq 15$ are computationally feasible. Note that the solution to the dynamic programming procedure will be a sequencing [1], [2], ..., [n] of the "jobs" and thus the optimal ordering (permutation) of the items will be the reverse sequence.

More recently, Shwimer [11] outlined a branch and bound approach to the general single machine sequencing problem. While the computational limitations of this approach are not encouraging (practical limits appear to be on the order of 20-25 jobs), "good" heuristic solutions should be readily available for larger sized problems.

Acknowledgement

We would like to thank Ms. Helen Cobb for writing the first version of our computer problem, while a student in the Industrial and Systems Engineering Department at the University of Florida.

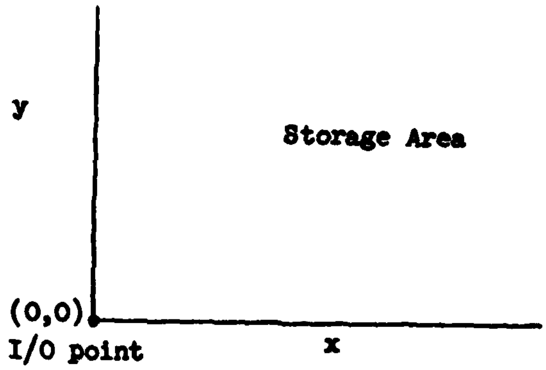


Figure 1. Warehouse Storage Area

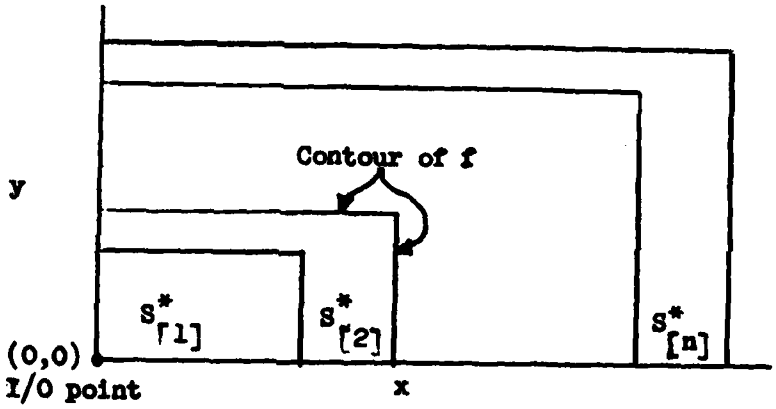


Figure 2. Optimal Storage Regions - Tchebychev Travel

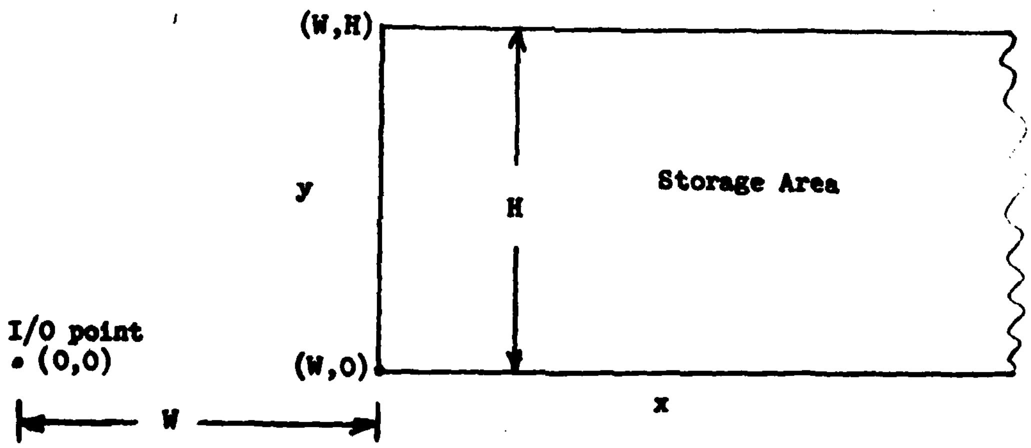
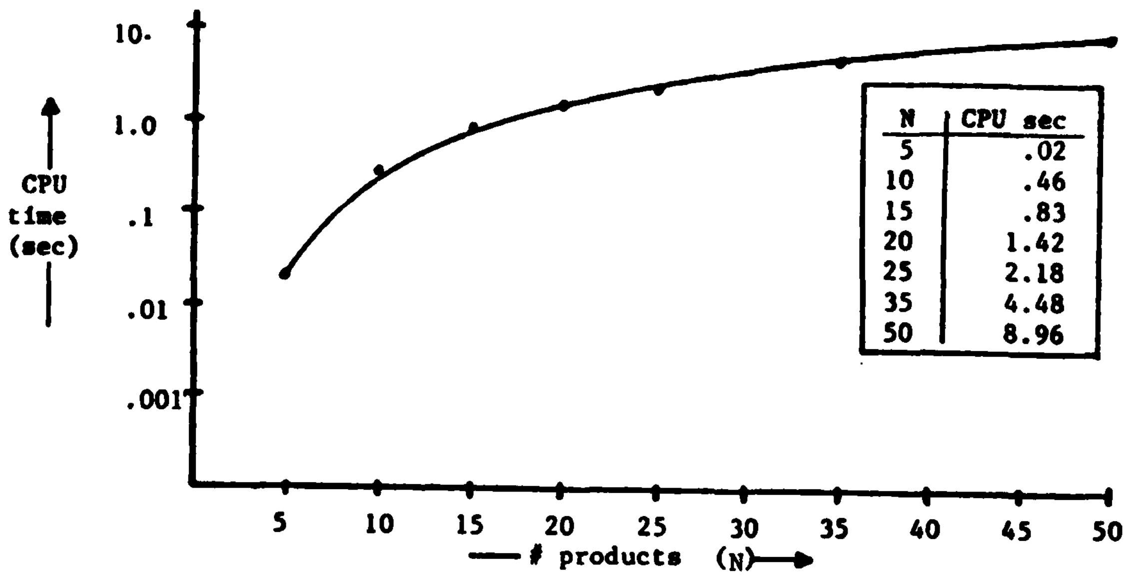


Figure 3. Travel Dominance Storage Area



N	CPU sec
5	.02
10	.46
15	.83
20	1.42
25	2.18
35	4.48
50	8.96

Table 1. Computation Time vs Number of Products

References

1. Avriel, Mordecai, Nonlinear Programming: Analysis and Methods, Prentice Hall, Inc., Englewood Cliffs, N.J., 1976.

2. Baker, Kenneth R., Introduction to Sequencing and Scheduling, John Wiley and Sons, Inc., New York, N.Y., 1974.

3. Francis, Richard L., and White, John A., Facility Layout and Location: An Analytical Approach, Prentice Hall, Inc., Englewood Cliffs, N.J., 1974.

4. Graves, Steven C., Hausman, Warren H., and Schwarz, Leroy B., "Storage-Retrieval Interleaving in Automatic Warehousing Systems," Management Science, Vol. 23, No. 9 (May 1977), pp. 935-945.

5. Heskett, James L., "Cube-Per-Order Index—A Key to Warehouse Stock Location," Transportation and Distribution Management, Vol. 4 (August 1964), pp. 23-30.

6. Hausman, Warren H., Schwarz, Leroy B., and Graves, Steven C., "Optimal Storage Assignment in Automatic Warehousing Systems," Management Science, Vol. 22, No. 6 (February 1976) pp. 629-638.

7. Kallina, Carl and Lynn, Jeffrey, "Application of the Cube-Per-Order Index Rule for Stock Location in a Distribution Warehouse," Interfaces, Vol. 7 (November 1976) pp. 37-46.

8. Lawler, Eugene L., "On Scheduling Problems with Deferral Costs," Management Science, Vol. 11, No. 2 (November 1964) pp. 280-288.

9. Luenberger, David G., Introduction to Linear and Nonlinear Programming, Addison-Wesley Publishing Company, Reading, Massachusetts, 1973.

10. Papineau, Robert L., Francis, Richard L., and Bartholdi, John J., "A Minimax Facility Layout Problem Involving Distances Between and Within Facilities," AIIE Transactions, Vol. 7, No. 4 (1975), pp. 345-355.

11. Shwimer, Joel, "On the N-Job, One-Machine Sequence-Independent Scheduling Problem with Tardiness Penalties: A Branch-Bound Solution," Management Science, Vol. 18, No. 6 (Febuary 1972) pp. B301-B313.

12. Thorton, V. Darryl, Francis, Richard L., and Lowe, Timothy J., "Rectangular Layout Problems with Worst-Case Distance Measures," AIIE Transactions, Vol. 11, No. 1, (March 1979) pp. 2-11.

13. Wilson, Hoyt G., "Order Quantity, Product Popularity, and the Location of Stock in a Warehouse," AIIE Transactions, Vol. 9, No. 3 (September 1977), pp. 230-236.

14. Zangwill, Willard I., Nonlinear Programming: A Unified Approach, Prentice-Hall, Inc. Englewood Cliffs, N.J. 1969.